

6. Diskreetsed automaadid

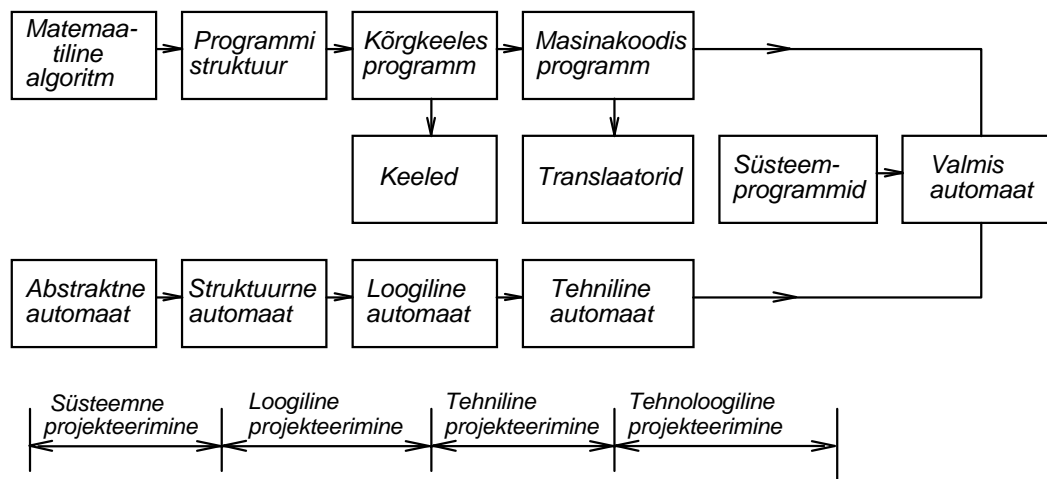
6.1. Diskreetsete automaatide olemus

Automaatide teooria põhineb diskreetsel olekuvõrranditel. Enamikul juhtimisobjektidel on lõplik arv diskreetsed olekuid. Seepärast nimetatakse nende juhtseadmeid lõplikeks automaatideks. **Lõplike automaatide teooria** on aluseks erinevate diskreetsete juhtseadmete, sealhulgas arvutite ja robotite juhtseadmete väljatöötamisel. Lõplike automaatide uurimiseks pole tingimata vaja füüsilise automaadi olemasolu. Kõige enam kasutatakse nende uurimiseks mitmesuguseid matemaatilisi mudeleid, mida nimetatakse **abstraktseteks automaatideks**. Kuna abstraktseid automaate saab kirjeldada algoritmikeelte abil, siis tuleneb sellest abstraktsete automaatide ning algoritmikeelte ekvivalentsus, s. t neid keeli on võimalik asendada abstraktsete automaatidega ja vastupidi. Üheks levinumaks ja kõige üldisemaks abstraktseks automaadiks on nn **Turingi masin**. Selle esitas 1936. a inglise loogik A M Turing. Masina tähtsus põhineb Turing-Churchi teesil, mille kohaselt igasuguse algoritmi infotöötuse võib sooritada Turingi masinaga. See väide ei ole matemaatiliselt tõestatav, sest algoritmi infotöötuse mõiste pole matemaatiline, vaid intuiitiivne. Katsed leida algoritmilisi protsesse kajastav formaalne eeskiri, mis oleks võimsam kui Turingi masin, on olnud edutud. Seepärast loetakse tänapäeval algoritmilisteks teisendusteks vaid teisendusi, mida saab teostada Turingi masinaga. **Abstraktne Turingi masin** on olnud aluseks arvutiteooria, programmeerimiskeelte ja keeletranslaatorite loomisel, ta on võimaldanud teoreetiliselt uurida juhtimisülesannete lahendamist, s. t määrata nende kuulumist algoritmiliste protsesside hulka jms. Universaalsed infotöötusseadmeteks on füüsilised arvutid, mille näol abstraktne Turingi masin on realiseerunud tegelikkuses. Järelikult on kõiki algoritmidega esitatavaid juhtimisprotsesse võimalik teostada arvutiga.

Automaatide teoorias vaadeldakse ka mitmeid lihtsamaid väiksema üldistusastmega abstraktsete automaatide mudeleid, mis võivad olla aluseks ühtede või teiste juhtseadmete väljatöötamisel. Niisuguste automaatide näiteks on **Mealy** ja **Moore'i automaadid**.

Diskreetsete automaatide projekteerimisel on neli etappi. Automaadi loomine algab süsteemse projekteerimisega, mille käigus koostatakse automaadi struktuur, leitakse põhilised plokid, määratakse nende otstarve ning funktsioonid. **Süsteemse projekteerimise** tulemuseks peab olema automaadi struktuuriskeem ning üksikute plokkide algoritmid. Seega minnakse abstraktselt automaadilt üle struktuursele automaadile. Järgmisel, **loogilise projekteerimise** etapil toimub automaadi loogiline süntees, tema sisemiste olekute minimeerimine ning automaadi ja selle plokkide funktsionaalskeemi koostamine. Üld- algoritmi järgi valitakse iga ploki ehitusviis ning loogikaelementide baas. Loogilisel projekteerimisel määratakse ka riist- ja tarkvara vahetõttu juhtseadmes. Loogiline projekteerimine peab andma automaadi kõikide plokkide funktsionaalskeemid ning nende struktuuri kirjeldused. Selleks kasutatakse automaatide loogiliseks sünteesiks ettenähtud erikeeli. Kolmandaks etapiks on automaadi **tehniline projekteerimine**, mille käigus lahendatakse kolm põhiülesannet: valitakse elemendid (integraallülitused), koostatakse nende paigutus- ja ühendusskeem trükiplaatidel ning trükiplaatide ja pistikühenduste paigutusskeem juhtseadme kapis. Erijuhul võib tehniline

projekteerimine tähendada ka spetsiaalse integraallülituse väljatöötamist. Tehnilise projekteerimise tulemuseks on juhtautomaadi tehniline dokumentatsioon, mille põhjal saab valmistada juhtautomaadi. Automaadi loomise viimaseks etapiks on **tehnoloogiline projekteerimine**. Selle käigus lahendatakse trükiplaatide valmistamise tehnoloogiaga, karkasside ja kappide valmistamisega, seadmete kontrolliga jms seotud probleeme. Tehnoloogilise projekteerimise tulemusel valmib dokumentatsioon, mille põhjal võib alustada juhtautomaatide saritootmist. Juhtautomaadi riist- ning tarkvara projekteerimise ja valmistamise etappidest annab ülevaate joonis 6.1.



Joonis 6.1. Juhtautomaadi projekteerimine ja valmistamine

Juhtautomaadi kirjeldamist on otstarbekas alustada abstraktse automaadi tundmaõppimisest. Automaati vaadeldakse kui musta kasti A (joonis 6.2), tema sisend- ja väljundsignaale aga kui tähestiku tähti. Kuna automaat on diskreetne, siis on abstraheritud ka aja kulg, mis võib omandada vaid diskreetseid järjestikuseid väärtusi, näiteks $t = 0, 1, 2, \dots$. Üldjuhul on niisugune automaat A kirjeldatav

- **sisendtähestikuga** ehk sisendsignaalide hulgaga

$$U = \{ u_0, u_1, \dots, u_i, \dots, u_l \}; \quad (6.1)$$

- **väljundtähestikuga** ehk väljundsignaalide hulgaga

$$Y = \{ y_0, y_1, \dots, y_j, \dots, y_m \}; \quad (6.2)$$

- **olekutähestikuga** ehk olekusignaali hulgaga

$$X = \{ x_0, x_1, \dots, x_k, \dots, x_n \}; \quad (6.3)$$

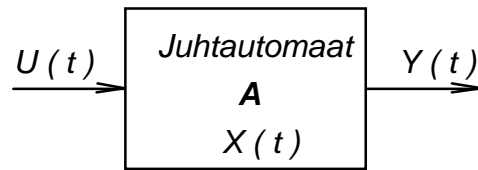
- **siirdefunktsiooniga**

$$S = f(x_k; u_i); \quad (6.4)$$

- **väljundfunktsiooniga**

$$V = f(x_k; u_i); \quad (6.5)$$

- automaadi **algolekuga** x_0 , mis vastab hetkele $t_0 = 0$.



Joonis 6.2. Abstraktne automaat

Siirde- ja väljundfunktsioonid määravad automaadi oleku $X(t+l)$ ja väljundsignaali $Y(t+l)$ hetkel $(t+l)$ sõltuvalt olekust $X(t)$ ja sisendsignaalist $U(t)$ hetkel t . Kui automaadi sisendite ja väljundite arv on suurem kui üks, võib signaali U , Y ja X vaadelda kui vektoreid või mitmetähelisi sõnu. Abstraktse automaadi töötamisel toimub sisendsõnade muutumine väljundsõnadeks, kusjuures protsessis etendab olulist osa automaadi sisemine olek antud hetkel. Iga järgmine olek oleneb eelmisest. Et väljundsignaalide ja olekute vahetumine toimuks soovitud korrapärasusega, tuleb automaadi mällu salvestada programm ning ette anda algolek hetkel $t = 0$. Kuigi kõigi diskreetsete automaatide olekud ja signaalid muutuvad diskreetsetel hetkedel, liigitatakse automaadid sõltuvalt nende ajalisest käitumisest sünkroonseteks ja asünkroonseteks. **Sünkroonsetes automaatides** toimivad sisendsignaalid täpselt fikseeritud hetkedel, automaadi sisemine olek muutub aga ajal, mil sisendsignaalide toime puudub. Ajahetkede fikseerimiseks on sünkroonsetes automaadis olemas taktiimpulsside generaator. **Asünkroonsetes automaatides** toimub olekute muutumine suvalistel hetkedel, mis on määratud ainult sisendsignaalide muutumisega. Automaadi olek mingil hetkel sõltub sellel samal hetkel saabuvatest sisendsignaalidest ning automaadi eelnenud olekutest. Kuna olekute muutumise hetked pole fikseeritud, siis puudub asünkroonsetes automaatides taktiimpulsside generaator. Mäluga automaatide väljundsignaalid sõltuvad nii sisendsignaalidest kui ka olekutest. Niisuguste automaatide hulka kuuluvad paljud seadmed, alates lihtsatest funktsionaalsetest loogikalülitustest nagu triggerid, loendurid ja registrid kuni mikroprotsessorite ja arvutiteni välja. Juhul kui automaadil puudub mälu, on tal ainult üks sisemine olek ning automaati kirjeldab täielikult väljundfunktsioon. Mäluta diskreetseid seadmeid nimetatakse **loogilisteks kombinatsiooniskeemideks**. Praktikas on sellisteks skeemideks dekodeerid, kommutaatorid, summaatorid jms.

Võib luua ka automaadi, millel sisendid puuduvad või mille sisendsignaalid ei muutu. Niisuguseid automaate nimetatakse **autonoomseteks**, kõiki ülejäänuid aga mitteautonoomseteks. Autonoomsete automaatide töö on täielikult määratud nende sisemiste olekutega e programmiga ning ei sõltu informatsioonist väljaspool automaati.

Peale loetletud automaatide on olemas veel mitmed spetsialiseeritud diskreetsete automaadid; neist tähtsam on **mikroprogrammautomaat**.

Mikroprogrammautomaadi idee esitas M V Wilkes 1951. aastal. Ta võttis selle kasutusele universaal-arvuti töö juhtimiseks. Kui tavaliselt arvuti ise juhib mingit protsessi, siis mikroprogrammautomaadi korral on tegemist automaadiga, mis juhib teist automaati, s. o arvutit ennast. Mikroprogrammjuhtimise põhimõtteid on rakendatud ka robotite ja tööpinkide juhtseadmetes.

Abstraktse automaadi matemaatilise kirjeldamise tulemuseks võivad olla: 1) algoritmide graaf- ja plokk skeemid, 2) siirde-, väljundi- ja olekutabelid, 3) loogikavõrrandid, 4) automaatide loogilise sünteesi algoritmikeelne kirjeldus või mõni teine esitusviis. Algoritmi praktiliseks teostamiseks saab kasutada kahte peamist võimalust: algoritmi realiseerimist **aparaadiks** või **programmiks**. Esimesel juhul on tulemuseks seade, teisel juhul programm. Seadme valmistamiseks kasutatakse mitmesuguseid pneumaatilisi, elektrilisi, optilisi või elektroonseid elemente. Sealjuures peab eelnevalt olema teada automaadi struktuur, mille aluseks on homogeenne või mittehomogeenne arvutuskeskkond (tehniline baas). Algoritmi programmilisel teostamisel tuleb koostada programm, mis on käskude jada, kusjuures need käsud määravad ära kõik juhtimiseks vajalikud operatsioonid, tehted lähteandmete ja vahetulemustega ning järgmise käsu aadressi. Programmi füüsiliseks kandjaks sobivad mitmesugused homogeened struktuurid (mälad). Programmi töötlemiseks: käskude lugemiseks, dešifreerimiseks ja täitmiseks kasutatakse mingit universaalset diskreetset automaati, näiteks mikroprotsessorit. Programmi koostamist nimetatakse algoritmi **programmiliseks modelleerimiseks**, selle aparatuurset realiseerimist aga algoritmi **aparatuurseks modelleerimiseks**.

6.2. Algoritmide riistvaraline realiseerimine

Automaatide aparatuurne realiseerimine algab automaadi matemaatilisest kirjeldusest, näiteks siirde- ja väljunditabelitest. **Siirde-** ja **väljunditabelid** erinevad kombinatsiooniskeemide olekutabelitest selle poolest, et nad kajastavad ka automaadi olekute ajalist muutumist. Siirdetabelist 1.9 näeb, milline on automaadi olek järgmisel taktil, kui on teada automaadi senine olek x ja sisendsignaali u . Uus olek leitakse sisendsignaali u_i ja olekusignaali x_k põhjal tabeli 1.9 vastavate veergude ja tulpade ristumiskohalt. Väljunditabel 1.10 võimaldab määrata automaadi väljundsignaali y_j , kui on teada tema olek ja sisendsignaali.

Tabel 6.1

Automaadi siirdetabel

$U_i \backslash X_k$	X0	X1	X2	X3	X4
U0	X1	X2		X3	X0
U1	X3	X0	X0	X4	
U2	X4		X3		X1

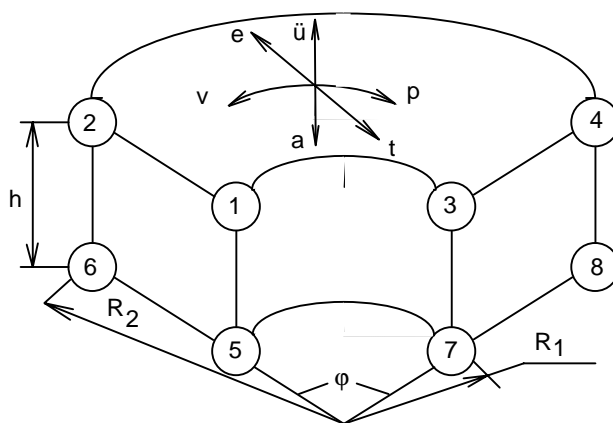
Tabel 6.2

Automaadi väljunditabel

$U_i \backslash X_k$	X0	X1	X2	X3	X4
U0	Y0	Y5		Y3	Y4
U1	Y2	Y0	Y4	Y1	
U2	Y3		Y2		Y0

Struktuurne automaat erineb abstraktsest selle poolest, et tal on teatud kindel arv sisendeid ja väljundeid, milles toimivad etteantud viisil kodeeritud sisend- ja väljundsignaalid. Lisaks sellele jagab struktuurne süntees automaadi kaheks - mäluks ja kombinatsiooniskeemiks. Automaadi mälu koosneb mäluelementidest. Viimastena saab kasutada kahe sisendiga *D*-trigereid, neist koostatud registreid, etteantud mälumahuga püsivõi muutmälusid, ketasmälusid jms mäluseadmeid.

Algoritmide aparatuursest realiseerimisest parema ettekujutuse saamiseks vaadeldgem struktuursete ja loogiliste automaatide sünteesi konkreetse näite varal. Selleks sobib tsüklilise positsioonjuhtimisega manipulaatori juhtautomaadi lihtsustatud mudel. Olgu tegemist silindrilises koordinaadistikus töötava manipulaatoriga, mida positsioonitakse teekonnalülite abil. Niisuguste manipulaatorite töötsoon ja võimalikud liikumised on näidatud joonisel 1.30. Positsioonimispunktid on tähistatud ringide ja vastava numbriga.



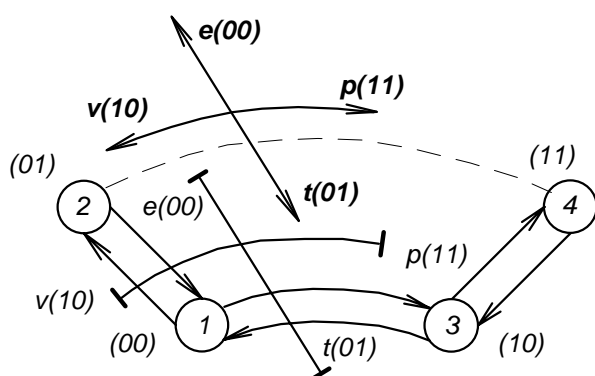
Joonis 6.3. Manipulaatori töötsoon, võimalikud liikumised ja positsioonimispunktid

Manipulaatori kolm lüli liiguvad kokku kuues suunas: edasi, tagasi, vasakule, paremale, üles ja alla, mida joonisel tähistavad nooled ja sõnade algustähed. Ajami liikumissuunda määrav käsk on juhtautomaadi väljundsignaaliks. Sisendsignaalideks on positsioonimispunktidesse paigutatud teekonnalülite signaalid. Iga võimalikku liikumist piiravad kaks teekonnalülitit: püstsihis liikumist all ja üleval, pöördliikumist paremal ja vasakul ning radiaalsuunalist liikumist ees ja taga asuvad lülitid. Seega on kokku kuus teekonnalülitit, mida tähistatakse tähtedega *ü* (ülemine lüliti), *a*, *v*, *p*, *e* ja *t*.

Manipulaatori positsioonimispunkte nimetatakse manipulaatori olekuteks, sest neile vastavad manipulaatori lülite eri asendid. Juhtseade peab võimaldama punktide vahel suvalist tsüklilist liikumist. Näiteks saab esemete teisaldamiseks kasutada tsükli 1, 2, 6, 2, 1, 3, 4, 8, 4, 3 ning edasi jälle 1, 2, 6, ... jne. Lihtsuse huvides ei ole siin arvestatud manipulaatori seiskamise ja haaratsi juhtimise käske. Oletatakse, et esemete haaramine ja vabastamine toimub positsioonimispunktis liikumissuuna muutumisel ilma peatumispausita.

Reaalne juhtseade peab väljastama käsked tehnoloogiaseadmetele ning võtma neilt vastu signaale. Seega on tegelikud juhtseadmed palju keerukamad kui siin vaadeldav juhtseade. Lihtsustused on vajalikud juhtautomaadi tööpõhimõtte paremaks mõistmiseks. Selguse huvides on otstarbekas näidet veelgi lihtsustada ning kirjeldada manipulaatori liikumist üksnes tasapinnal. Joonisel 1.31 on kasutatud samu tähiseid mis ruumilise liikumise mudeli juures. Lisaks on toodud väljundsignaalide, olekute ja sisendsignaalide koodid. Väljundsignaalide koodid on tähistatud rasvaselt, kusjuures vastavaid liikumisi näitavad nooled. Sisendeid tähistavad liikumise piirikud ning nende kõrval on näidatud koodid. Oletagem, et manipulaator töötab tsükliga 1, 2, 1, 3, 4, 3 ning edasi jälle 1, 2, 1, ... jne.

Manipulaatori liikumist kirjeldavat skeemi võib vaadelda kõigi eespool loetletud lihtsustuste korral ka juhtautomaadi graafskeemina, mis määrab roboti töö algoritmi. Graafskeemi järgi saab koostada juhtautomaadi siirde- ja väljunditabelid (tabelid 1.11 ja 1.12). Arusaadavuse huvides on tabelis nooltega näidatud roboti töösükkel. Siirdetabelis esitatud automaadi olekud vastavad tema mälu olekutele. Kokku on tabelisse kantud nelja liiki signaalide koodid. Need on sisend- ja väljundsignaalide ning mälu sisendi ja väljundi ehk eelmise ja järgmise oleku koodid. Viimaseid tähistatakse tähtedega D ja M (joonis 1.32).



Joonis 6.4. Manipulaatori töösükli graafskeem

Kõik koodid on kahekohalised. Nende nooremat ehk parempoolset kohta tähistab vastavalt u_0, y_0, D_0 või M_0 ; vanemat ehk vasakpoolset kohta aga u_1, y_1, D_1 või M_1 . Tühjad lahtrid tabelis on tingitud sellest, et vaadeldav liikumine ei hõlma manipulaatori kõiki võimalikke liikumisi ning alati pole ka kõigile tabeli lahtritele vastavad tegevused võimalikud.

Tabel 6.3

Automaadi siirdetabel

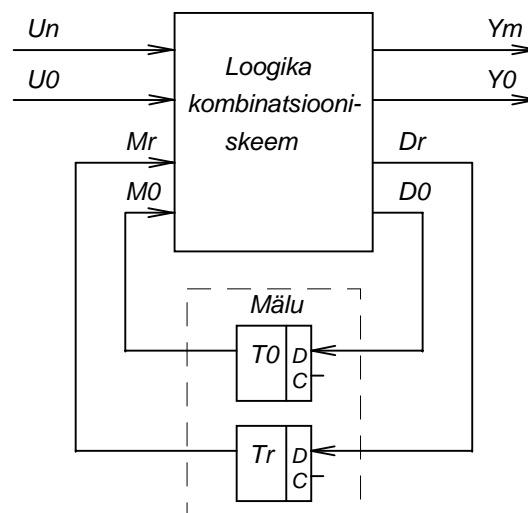
		X_{k+1}			
		M_1 M_0	01	10	11
u_i	$X_k (M)$				
	u_1 u_0				
	01				
	10				
	11				

Diagrammatic representation of the transition table with arrows and state labels: D_1, D_0 for the first column; $00, 01, 10, 11$ for the other columns.

Tabel 6.4

Automaadi väljunditabel

		Y_j			
		$M_1 M_0$ 00	01	10	11
u_i	$X_k (M)$				
	$u_1 u_0$ 00		01		01
	01	11		10	
	10	00			
	11			00	



Joonis 6.5. Struktuurne automaat

Manipulaatori liikumise, s. o juhtautomaadi programmeerimiseks tuleb tema olekud salvestada mälu. Kuna olekut kirjeldab kahekohaline kahendarv, siis piisab mälest, mis koosneb kahest D -trigerist (joonis 6.5). Automaadi loogiline süntees seisneb tema loogika kombinatsiooniskeemi väljatöötamises. Selleks on vaja automaadi siirde- ja väljunditabelite põhjal koostada automaadi tööd kirjeldavad loogikafunktsioonid. Nende funktsioonide all mõeldakse väljundsignaale y_0, y_1, D_0 ja D_1 , mis on sisendite u_0, u_1, M_0 ja M_1 funktsioonid. Püstitatud ülesande võib lahendada kahel viisil. Esiteks võib luua universaalse loogilise automaadi, mis peale ühenduste tegemist on võimeline sooritama ükskõik millist antud tabelis kirjeldatud tegevust. Ainsaks tingimuseks on, et signaalide arv ja tabelite mõõtmed ei muutuks. Sel juhul on tegemist automaadi aparatuurse liiasusega, mis võimaldab automaati vajaduse korral ümber häälestada. Teiseks võib koostada loogikaskeemi, mille korral automaat täidab ainult üht konkreetset tabelis näidatud

tegevust. Sel juhul on tegemist eriotstarbelise automaadiga, mille tööd pole võimalik muuta.

Loogikafunktsioonide esitamiseks on otstarbekas siirde- ja väljunditabelist üle minna olekutabelitesse. Lihtsamal juhul võib need funktsioonid avaldada ka vahetult siirde- ja väljunditabelitest. Olekutabelisse koondatakse kõik sisendsignaalide ning neile vastavate väljundsignaalide väärtused. Tabelite 6.3 ja 6.4 põhjal koostatud automaadi olekutabelis (tabel 6.5) on näiteks neli sisendsignaali ja neli väljundsignaali. Sisendsignaalid võivad esinevad 16 kombinatsioonis, kuigi eespool toodud näite puhul oli tegemist vaid 6 kombinatsiooniga.

Tabel 6.5

Automaadi olekutabel

Sisendid				Väljundid			
U1	U0	M1	M0	D1	D0	Y1	Y0
0	0	0	0				
0	0	0	1				
0	0	1	0				
0	0	1	1				
0	1	0	0				
0	1	0	1				
0	1	1	0				
0	1	1	1				
1	0	0	0				
1	0	0	1				
1	0	1	0				
1	0	1	1				
1	1	0	0				
1	1	0	1				
1	1	1	0				
1	1	1	1				

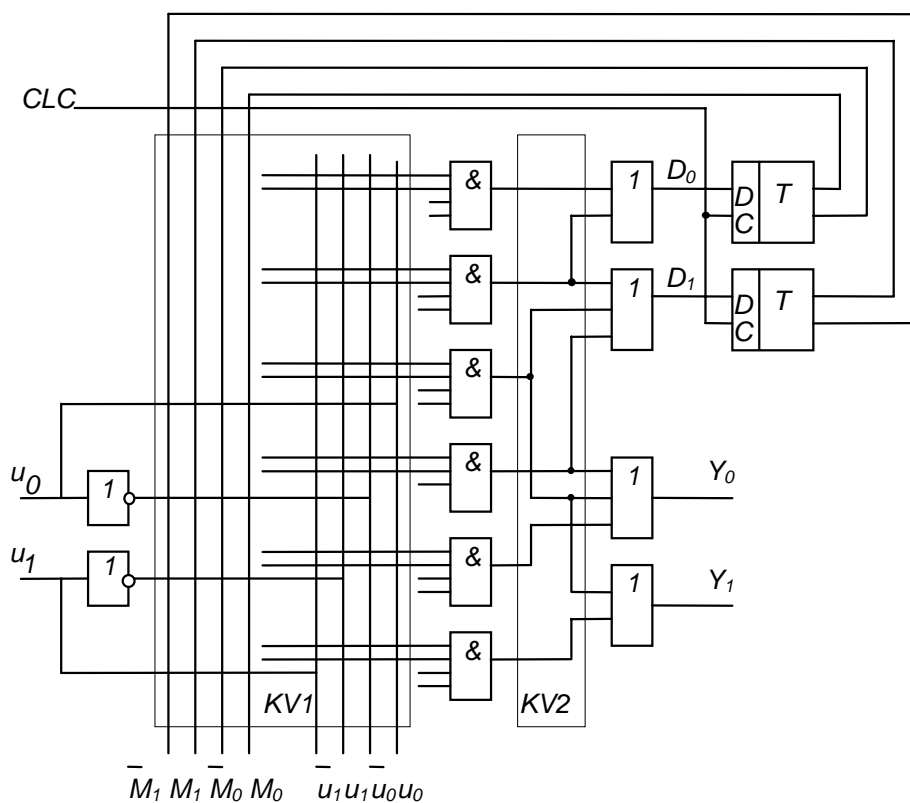
Olekutabeli 6.5 järgi kirjutatakse välja automaadi loogikavõrrandid, kusjuures arvesse võetakse ainult need sisendsignaalide kombinatsioonid, mille korral väljundsignaal võrdub ühega. Nii saadakse alljärgneval kujul esitatud loogikavõrrandid (puuduvad loogilise eituse märgid)

$$\begin{aligned}
 D_1 &= u_1u_0M_1M_0 \vee u_1u_0M_1M_0 \vee u_1u_0M_1M_0, \\
 D_0 &= u_1u_0M_1M_0 \vee u_1u_0M_1M_0 \vee u_1u_0M_1M_0, \\
 Y_1 &= u_1u_0M_1M_0 \vee u_1u_0M_1M_0 \vee u_1u_0M_1M_0, \\
 Y_0 &= u_1u_0M_1M_0 \vee u_1u_0M_1M_0 \vee u_1u_0M_1M_0,
 \end{aligned}
 \tag{6.6}$$

Võrrandite (6.6) põhjal saab konstrueerida juhtautomaadi loogikaskeemi. Et skeem kujuneks võimalikult lihtsaks, tuleb loogikafunktsioonid minimeerida. Võrranditest on

näha, et eri funktsioonid sisaldavad ühesuguseid loogilisi osakorrutisi, mis lihtsustab oluliselt juhtautomaadi loogikat.

Nendega on võimalik tutvuda ka kirjanduse abil. Oluline on, et kui tahetakse luua roboti juhtseade, mis juhiks manipulaatori töötsiooni ulatuses kõiki täiturseadme mehhanismide ja ajamite võimalikke liikumisi, siis peab juhtautomaat sisaldama vahendeid sisendsignaali suvalistele kombinatsioonidele vastavate loogikafunktsioonide realiseerimiseks. Antud juhul peab niisugune automaat koosnema vähemalt 16 loogilisest NING-elementist ja neljast VÕI-elementist. Igal NING-elementil olgu vähemalt 4 sisendit ja igal VÕI-elementil kuni 8 sisendit. Lisaks vajatakse signaalide inverteerimiseks EI-elemente. Kui automaadiga soovitakse realiseerida ainult ühte algoritmi näiteks tabelitega 6.3 ... 6.5 esitatut, siis automaadi loogikaskeem lihtsustub ning piisab kuuest NING- ja neljast VÕI-elementist. Sellise automaadi loogikaskeemi struktuur on joonisel 6.6.



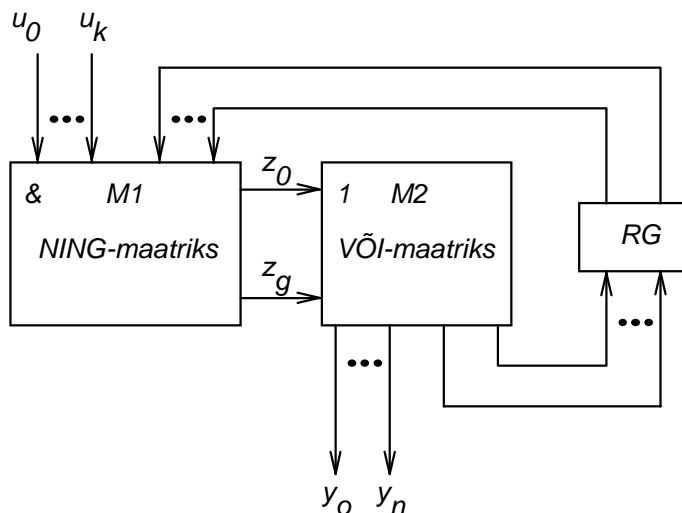
Joonis 6.6. Juhtautomaadi loogikaskeem

Automaadil on kaks sisendit ja kaks väljundit. Manipulaatori juures kasutatakse aga nelja täiturit, näiteks kontakteid, mis lülitavad ajameid liikuma edasi - tagasi või paremale ja vasakule. Kasutatakse ka nelja teekonnalülitit. Et ühendada juhtseade manipulaatoriga, tuleb sisendsignaalid kodeerida ning väljundsignaalid dekodeerida. Täiturite juhtimiseks on neid vaja võimendada. Kuna mälulementideks on sünkroonsed D-trigerid, vajatakse automaadi tööks taktiimpulssse. Viimased saadakse taktiimpulsside generaatorist. Skeemil

on kasutatud NING- ja VÕI-elemente. Kuna integraallülitustena toodetakse peamiselt NING-EI- ja VÕI-EI-elemente, siis võib automaadi tegelik skeem joonisel 1.33 esitatust veidi erineda.

Loogikaelementide sisendites olevaid kommutatsioonivälju $KV1$ ja $KV2$ saab valmistada ümberhäälestatavatena, kus ühendused tehakse näiteks pistikute abil. Ühendusskeemi muutmisel muutub juhtautomaadi töö algoritm. Sama otstarvet täidab ka ümberprogrammeeritav loogiline maatriks. Automaadi häälestamist võib tinglikult nimetada programmeerimiseks. Vaadeldud põhimõttel töötavad lihtsate tsükliliste positsioonjuhtimisega robotite ja muutumatu töösükliga autooperaatorite juhtseadmed. Antud skeem on koostatud ühe konkreetse algoritmi jaoks.

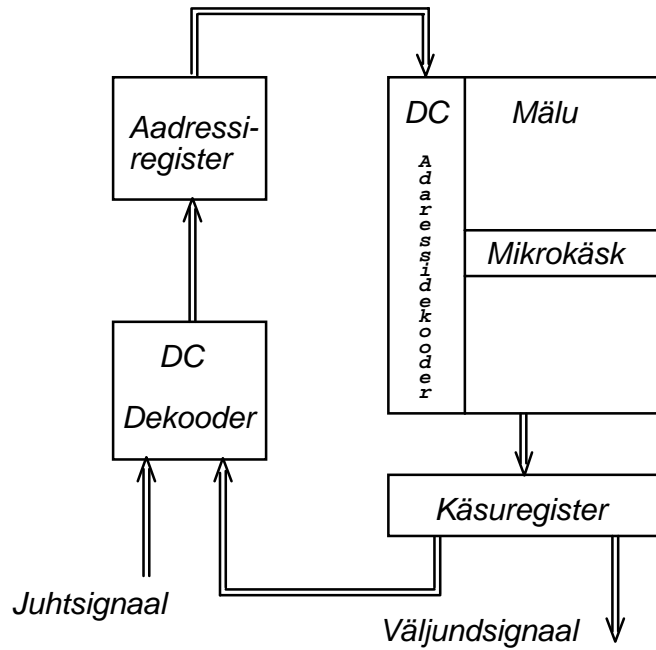
Maatriksitel põhinevat loogikalülitust saab hõlpsasti muuta juhtautomaadiks, kui lisada mäluregister. Niisuguse juhtautomaadi struktuur on joonisel 1.34. Ka see, et juhtautomaate võib lülitada järjestikku, kinnitab maatriksitega juhtautomaatide universaalsust. Keeruline loogikafunktsioon realiseeritakse sel juhul mitmeastmeliselt, kasutades vahemuutujaid. Selliseid automaate kasutatakse sageli keerukate juhtseadmete sisemiste protsesside mikroprogrammjuhtimiseks.



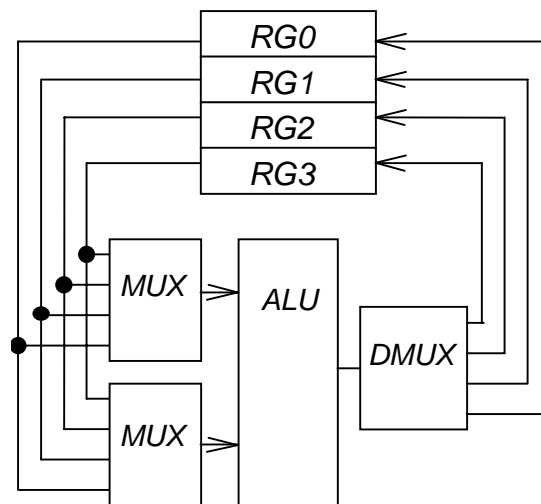
Joonis 6.7. Programmeeritavate maatriksitega realiseeritud automaat

Programme saab salvestada ka pooljuhtmällu. Mälul põhineva programmautomaadi struktuur on näidatud joonisel 6.8. Mikrokäsu aadress salvestatakse aadressi- registrisse. Aadressi dešifreerimisel leitakse mälus sellele vastav mikrokäsk, mis sisaldab järgmise mikrokäsu aadressikoodi ja antud mikrokäsule vastava mikrooperatsiooni koodi. Mikrokäsk loetakse mälust käsuregistrisse. Sealt edasi läheb aadressikood dešifraatorisse DC. Aadressikoodi ja väljast saabuva juhtkäsu alusel moodustatakse dešifraatoris sünkroimpulsi saabumisel järgmise mikrokäsu aadress, mis salvestatakse aadressiregistrisse. Käsuregistrist väljastatakse seadme juhtimiseks vajalik mikrooperatsiooni kood. Kuigi programmautomaate realiseeritakse nii mälude kui ka

programmeeritavate loogiliste maatriksite baasil, on eri automaatidel veidi isesugused omadused. Mälumassiivi dešifraator on jäiga loogikaga, *PLM*- maatriks *MI* aga pindliku loogikaga. *PLM* realiseerib loogikafunktsiooni $y_i = f(x_i)$ disjunktiiivset normaalkuju, mälu aga selle funktsiooni tõeväärtustabelit. Sellest tingituna on mäludega lihtsam realiseerida lihtsa loogika ja suure väljundite arvuga funktsioone. *PLM*-i on otstarbekas kasutada seal, kus on tegemist suhteliselt keerukate loogikafunktsioonidega, ning juhul kui väljundite arv ei ole väga suur.



Joonis 6.8. Mälul põhinev programmautomaat



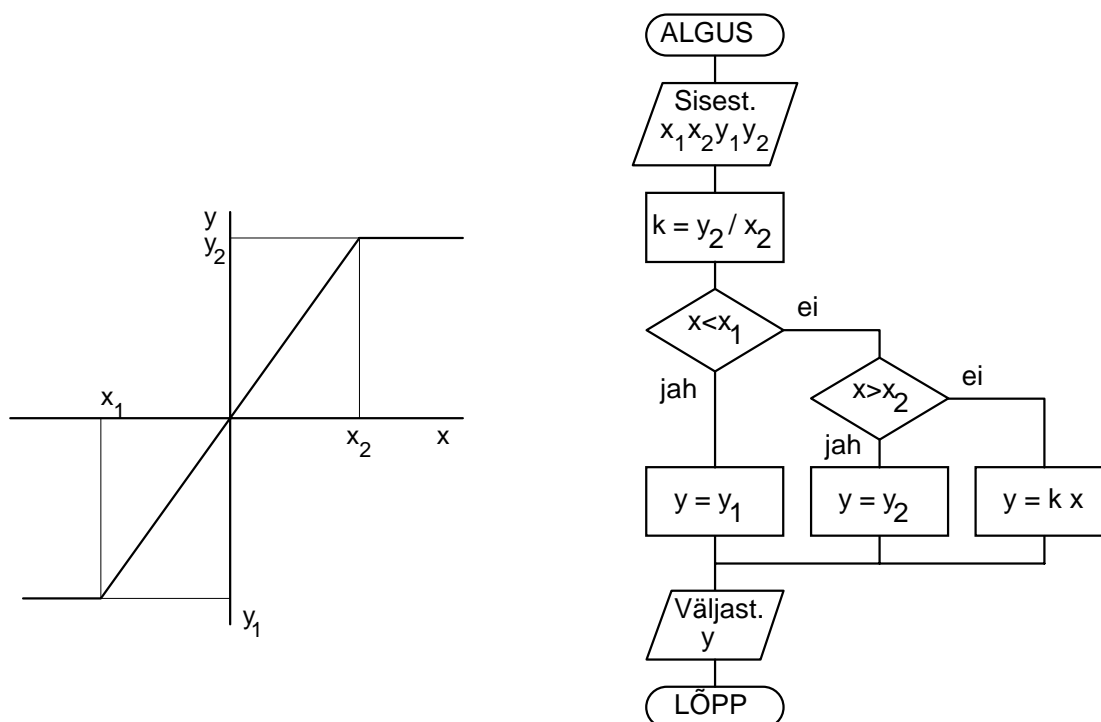
Joonis 6.9. Aritmeetika-loogikaploki kasutamine koos kommutaatorite ja registritega

6.3. Algoritmide tarkvaraline realiseerimine

Juhtseadmete riistvara ja tarkvara duaalsusest (ühildatavusest ja asendatavusest) tuleneb, et juhtalgoritme saab realiseerida nii aparatuursete (riistvara) kui ka programmiliste (tarkvara) vahenditega. Esimesel juhul on realisatsiooni tulemuseks **aparatuurne automaat**, teisel juhul **programmiautomaat**. Programmiautomaat kujutab endast mõnes algoritmikeeles esitatud programmi, mis on ette nähtud protsessi või seadme juhtimiseks. Programmi salvestamiseks ja töötlemiseks läheb aga vaja universaalset riistvaraautomaati. Tänapäeval on niisugusteks automaatideks mikroprotsessoritel põhinevad üldotstarbelised arvutid, raalid ja raalisüsteemid. Juhtraalide mälu maht, töökiirus, sisendite ja väljundite arv, väline kuju ning hind on väga erinevad. Kasutatakse nii odavaid binaarsüsteemide juhtimiseks mõeldud programmeeritavaid kontrollereid kui ka keerukaid, hierarhilise või hajasstruktuuriga mitmeraali juhtseadmeid. Vaatamata mikroprotsessorsüsteemide mitmekesisusele on eri tüüpi protsessorite ehituses ja tööpõhimõttes palju sarnast.

Algoritmi plokk skeemid

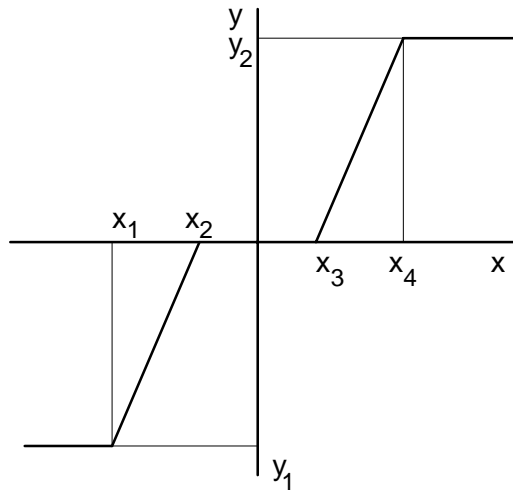
Mittelineaarsete karakteristikute modelleerimine. Analoogsüsteemides kasutatakse süsteemi muutujate piiramiseks mittelineaarse karakteristikuga regulaatoreid. Tüüpilisteks mittelineaarsuseks on "tundetustsoon" ja "küllastus". Analoogiliste piirangute loomine on võimalik ka arvregulaatorites. Joonisel 4. on toodud tundetustsooni ja küllastusega funktsiooni algoritm. Kuna tegemist on tükiti lineaarse funktsiooniga, siis tuleb programmeerimisel ette anda selle üksikute lõikude võrrandid ning käänupunktide koordinaadid.



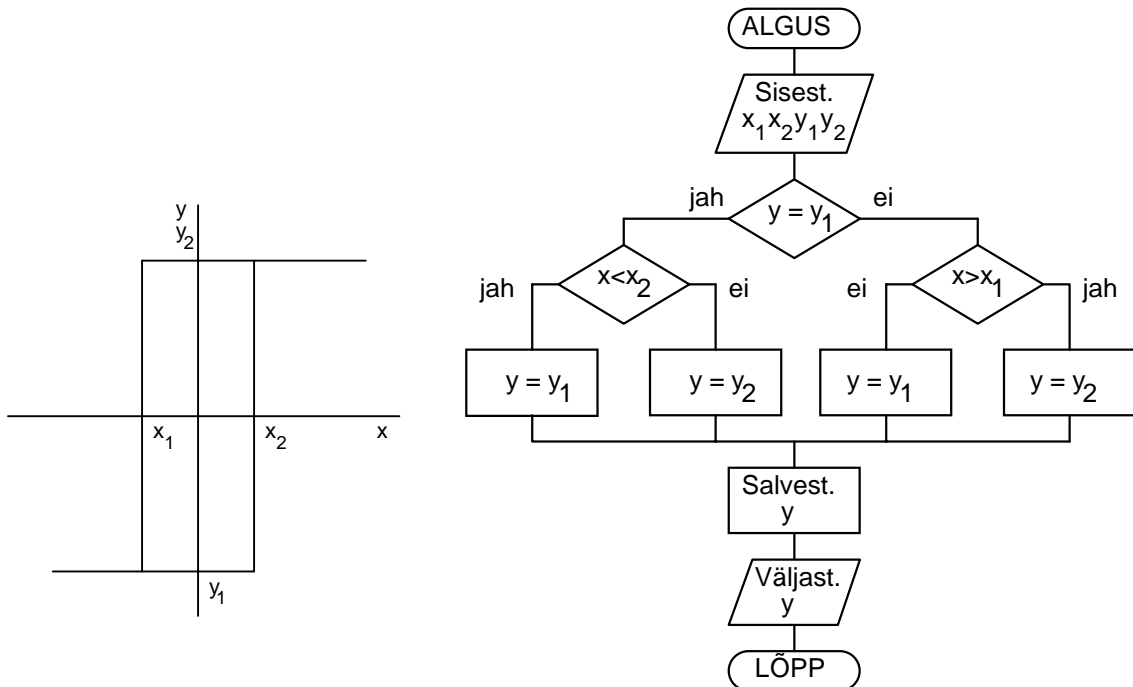
Joonis 6.10. Küllastustsooniga mittelineaarne karakteristik ja sellele vastav

modelleerimise algoritm

Arvregulaatoris realiseeritakse piirangute algoritm alamprogrammina, mis täidetakse pärast regulaatori väljundfunktsiooni programmi. Sarnaste piirangute korral võib piirangute alamprogrammi kasutada korduvalt erinevate regulaatorite juures, muutes seejuures vaid piirangfunktsiooni parameetrite arvvaartusi.



Joonis 6.11. Tundetuse- ja küllastustsooniga mittelineaarne karakteristik



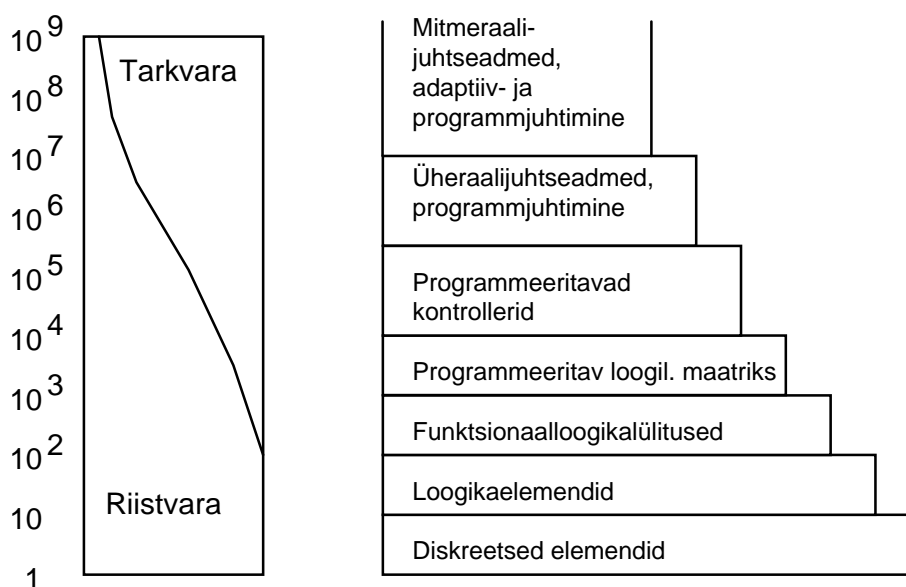
Joonis 6.12. Tükiti lineaarne hüsteresifunktsioon ning algoritmi plokk skeem selle modelleerimise

6.4 Riist- ja tarkvara hierarhia

Roboti juhtseadmed täidavad mitmesuguseid juhtimisfunktsioone. Sõltuvalt juhtimise eesmärgist ning juhtimisobjekti mudelist koostatakse süsteemi projekteerimisel juhtseadme talitluse eeskiri ehk tegevusjuhise, mida nimetatakse **juhtimise algoritmiks**. Algoritmi realiseerimiseks on kaks teed. Esiteks võib konstrueerida spetsiaalse juhtseadme, mille elemendid ja struktuur määravad üheselt ära seadme talitluse. Seega kätkeb seadme aparaatuur ehk **riistvara** endas ka juhtimise algoritmi. Teiseks võib kasutada universaalset juhtseadet, mille talitluse määrab ära salvestatud **programm**. Universaalseteks juhtseadmeteks on arvutid, mis võimaldavad realiseerida mitmesuguseid formaliseeritud tegevusjuhiseid. Arvuti või juhtseadme tööks vajalikku programmide kogumit nimetatakse seadme programmvaraks ehk **tarkvaraks**. Juhtseadme riistvara ja tarkvara on lahutatud. See tähendab, et riistvara on vajalik üksnes siis, kui ta täidab mingit algoritmiga etteantud juhtimisfunktsiooni. Ja vastupidi, tarkvara saab täita oma funktsioone ainult siis, kui on olemas riistvara, mille abil need funktsioonid realiseeritakse. Selles väljendub **riistvara ja tarkvara dualism**. Juhtseadme konkreetse kasutaja seisukohast vaadatuna võib suurema tähtsusega olla kord seadme riistvara, kord tarkvara.

Juhtseadme riist- ja tarkvara koosneb elementidest ja plokkidest, mida ühendab kindel struktuur. Enamiku juhtseadmete riistvara põhineb pooljuhtelementidel, alates transistoridest ja diodidest kuni suurte integraallülitusteni välja. Sõltuvalt integratsiooniastmest ja funktsionaalsetest võimalustest on need elemendid paigutatud joonisel 1.40 eri hierarhiatasanditele. Nendest koostatud juhtseadmed erinevad üksteisest ka keerukuse ja funktsionaalsete võimaluste poolest. Robotite juhtseadmetest on lihtsaimad programmeeritavad kontrollid. Täiuslikemate robotite juhtimiseks kasutatakse ühe- ja mitmeraali juhtseadmeid.

Elementide arv
(integratsiooniaste)



Joonis 6.13. Diskreetsete juhtseadmete hierarhia

Tarkvara elementideks on mitmesugused programmid. Programmidel on samuti hierarhiline struktuur, kus ühed programmid on teiste suhtes alamprogrammideks, teised omakorda aga kolmandate alamprogrammideks.

Hierarhiat loetakse süsteemi keerukuse tunnuseks. Teatud keerukusest alates ei suuda üks keskjuhtseade enam kõiki juhtimisfunktsioone rahuldavalt täita. Siis võetakse üheraalijuhtseadmete asemel kasutusele mitmeraali juhtseadmed ning hierarhilise juhtimise põhimõtte. Kogu juhtimissüsteem muutub sel juhul mitmetasandiliseks. Juhtimisfunktsioonid jaotatakse eri tasandite vahel: alumine tasand täidab suhteliselt lihtsaid, kuid kiiret reageerimist nõudvaid ülesandeid, ülemised aga keerukamaid, üldisema iseloomuga ülesandeid. Kahetasandilise süsteemi korral nimetatakse eri tasanditel toimuvat juhtimist vastavalt **lokaalseks** ja **keskjuhtmiseks**. Roboti puhul juhitakse lokaalsel tasandil manipulaatori ajameid, kusjuures iga ajamit juhib eraldi juhtplokk. Keskjuhtimine on ette nähtud manipulaatori lülide liikumise kooskõlastamiseks ning roboti tööprotsessi korraldamiseks.

Mitmeraali juhtseadmetes jaguneb ka tarkvara eri juhtimistasandite vahel. Seejuures moodustab tarkvara riistvarast sõltumatu hierarhilise süsteemi, mis täiendab kogu juhtimissüsteemi uute tasanditega. Näiteks võib tarkvara abil lisaks programmijuhtimisele luua adaptiivjuhtimise või tehisintellekti tasandid, mille ülesandeks on programmide automaatne muutmine või uute programmide koostamine.

Juhtseadmed võivad olla spetsiaalsed või universaalsed. Esimesed on ette nähtud ühetüübiliste objektide ühetaoliseks juhtimiseks. Teised võimaldavad juhtida eri objekte (protsesse ja seadmeid). Nii juhtseadmete riist- kui ka tarkvara liigitatakse spetsiaalseks ja universaalseks. Seadmed, juhtplokid ja mehhanismid, mille konstruktsioon määrab üheselt nende funktsioonid, on **spetsiaalne ehk eririistvara**. Sellele vastandina on **universaalne riistvara** mitmeotstarbeline ja paindliku konstruktsiooniga. Universaalsus saavutatakse mehhanismide ümberseadistamise, juhtseadmete ümberhäälestamise või ümberprogrammeerimise abil. Mainitud tegevust võib nimetada **unifitseerimiseks**. Selle abil saab suhteliselt lihtsalt anda seadmele uue otstarbe. Võrreldes unifitseerimise viise omavahel, osutub, et kõige tülikam on mehhanisme ümber seadistada, kõige hõlpsam aga juhtseadmeid ümber programmeerida. Seepärast kasutatakse universaalsete automaatide juhtimiseks ümberprogrammeeritavaid juhtseadmeid.

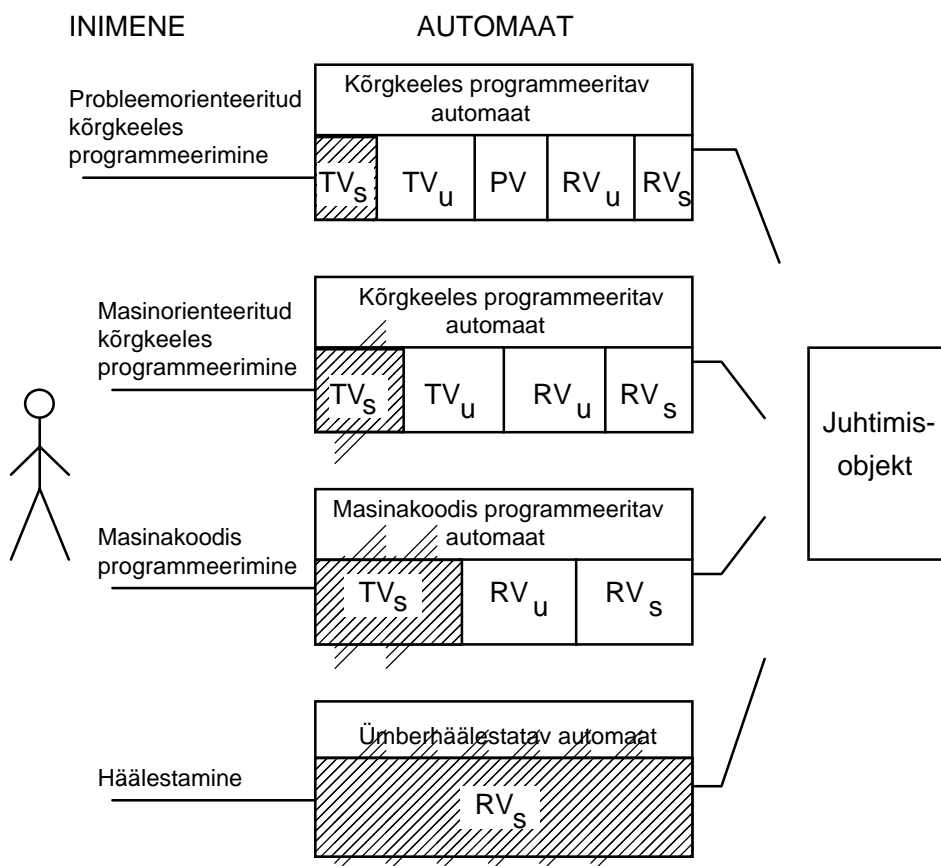
Programmide salvestamiseks, säilitamiseks ning programmijuhtimiseks on **mälu**, kuhu võib kirjutada erisuguseid programme. Mäluseadet loetakse samuti universaalseks riistvaraks, kusjuures mälu maht on juhtseadme universaalsuse mõõt.

Programmide koostamist ja salvestamist nimetatakse **programmeerimiseks**. Inimese jaoks tähendab see suhtlemist automaadiga (arvuti või robotiga). Programmeerida on seda lihtsam, mida mõistetavamad on programmides kasutatavate käskude sümboolid ning mida arusaadavam on programmi struktuur. Kõige lihtsam oleks inimkeelne kirjeldus. Kahjuks pole inimkeel alati üheselt mõistetav ja masinale arusaadav. Seepärast kasutatakse juhtseadmete programmeerimiseks **algoritmikeeli**. **Assemblerkeeled** on koostatud juhtseadme või arvuti riistvara ehitusest lähtudes. Nende kasutamine nõuab programmeerijalt

juhtseadme talitluse üksikasjalikku tundmist. Niinimetatud **kõrgkeeled** on inimlähedasemad. Kõrgkeele kasutaja ei pea tundma juhtseadme ehitust. Programmeerimisel tuleb kirjeldada robotilt nõutavaid liikumisi või roboti tööülesannet.

Robotite kõrgkeeli liigitatakse nt. **masin-** ja **probleemorienteeritud** keelteks. Neist esimesed on kasutusel kindlat tüüpi robotite korral. masinorienteeritud keelteks nimetatakse neid sellepärast, et programmis kirjeldatakse manipulaatori üksikuid liikumisi, mitte aga tööoperatsioone, milleks liikumised on vajalikud. Probleemorienteeritud keeled on universaalsemad. Vastav programm kirjeldab tehnoloogiaprotsessis nõutavaid operatsioone, mitte roboti käe liigutusi.

Programme salvestatakse ja töödeldakse kahendkoodis, mida nimetatakse ka **masinakoodiks** Inimesele on see raskesti jälgitav ja väheülevaatlik. Inimesel on masinakoodis programmi koostada väga tülikas. Et teisendada inimese koostatud kõrgkeelne programm masinale arusaadavasse kahendkoodi, peab juhtseade kasutama teatud sisemist ehk **süsteemset tarkvara**. Kuna selle tarkvara sisu ei sõltu roboti konkreetsest tööülesandest, nimetatakse seda ka **universaalseks tarkvaraks**. Robotsüsteemi juhtimiseks ettenähtud konkreetset tööprogrammi võib lugeda aga **spetsiaalseks** ehk **eritarkvaraks**.



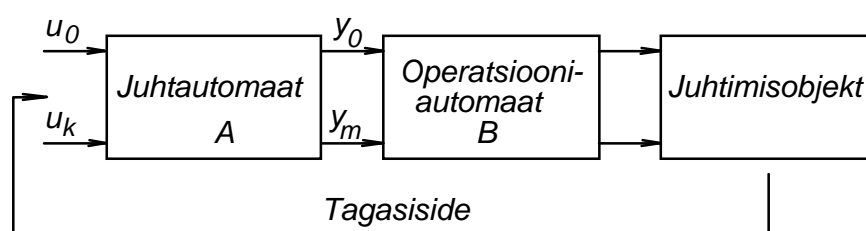
Joonis 6.14. Automaadi riist- ja tarkvara ning inimese suhe automaadiga

Tarkvara moodustab juhtseadme riistvara ja inimese vahel hierarhilise (mitmekihilise) süsteemi. Osa tarkvarast on vahetus kokkupuutes riistvaraga, osa inimesega. On otstarbekas, et inimene täidab ainult neid spetsiifilisi ülesandeid, millega juhtseade toime ei tule. Niisuguseks ülesandeks on näiteks robotile konkreetse tööprogrammi koostamine. Mida täiuslikum on robot, seda mahukam on tema mällu salvestatud universaalne tarkvara, mida robot vajaduse korral ilma inimese abita kasutab. Universaalse riistvara ja tarkvara ühitamise tulemusena on loodud niinimetatud **püsivara**, mis tähendab püsimalukiipidesse salvestatud programme.

Inimese ja automaadi ning riist- ja tarkvara vahekordi näitab joonis 1.41, kus RV tähistab riistvara, TV tarkvara ja PV püsivara. Indeksid u ja s tähistavad riistvara ja tarkvara universaalseid ja spetsiaalseid osasid. Mida universaalsem on automaat, seda vähem peab inimene kohandama teda uueks tööks. Joonisel näitab seda viirutatud alade pindala.

Automaatjuhtimise võimalusi on aastakümneid piiranud juhtseadmete riistvara tehniline tase. Arvutustehnika tormiline areng on olukorda juhtimistehnikas muutnud. Raalid on universaalsed juhtseadmed, mille riistvara ja tarkvara võimaldavad realiseerida väga keerukaid juhtimisalgoritme. Seepärast on juhtimisel kujunenud määravaks nõutavate programmide ehk tarkvara loomine.

Süsteemide keerukuse teatud tasemel on juhtimise lihtsustamiseks otstarbekas rakendada hierarhilist struktuuri. Vastavalt sellele võib keeruka automaadi jagada juht- ja operatsiooniautomaadiks. Niisuguse automaadi struktuur koos juhtimisobjektiga on näidatud joonisel 1.42. Operatsiooniautomaat väljastab käsusignaale manipulaatorile ja tehnoloogiaseadmetele. Juhtautomaat korraldab operatsiooniautomaadi tööd. Arvutis on operatsiooniautomaadiks protsessor, juhtautomaadiks aga protsessori tööd juhtiv mikroprogrammautomaat. Operatsiooniautomaadi iseärasuseks on suur väljundsignaalide arv. Juhtautomaadil tuleb lahendada keerukaid loogikaülesandeid.



Joonis 6.15. Keerulise automaadi jagamine juht- ja operatsiooniautomaadiks

Näiteks, robotsüsteem töötab kaheastmelise juhtimise seisukohalt järgmiselt. Süsteemi töötükk koosneb üksikutest tööoperatsioonidest, nagu manipulaatori lülide liikumine ühest positsioonimispunktist teise, detaili haaramine või vabastamine, tööpingi sisselülitamine jne. Iga taolise näiliselt lihtsa operatsiooni täitmiseks peab roboti juhtseade tegema rohkesti elementaartehteid: sisendsignaalist sõltuvalt vastu võtma loogilisi otsuseid, leidma etteantud positsioonimispunkti ja tegeliku asendi erinevuse, moodustama juhtimis- ja indikatsioonisignaale. Juhtseadme töös tähendab see mitmesuguste aritmeetika- ja loogikaülesannete lahendamist ning seadmete (aritmeetika-loogikaploki

registrite) üksikasjalikku juhtimist. Seega tuleb roboti iga tööoperatsiooni jaoks täita juhtimisalgoritm, mis sisaldab sadu elementaartehteid. Roboti juhtimine ja programmeerimine elementaartehte kaupa on äärmiselt ebamugav ning aeganõudev. Peale selle vajatakse väga kõrge kvalifikatsiooniga spetsialiste, kes tunnevad põhjalikult seadme ehitust. Hoopis mugavam on koostada robotsüsteemi töö algoritm manipulaatori ja tehnoloogiaseadmete üksikute tööoperatsioonide kaupa. Sel juhul peavad aga juhtseadmetes olema realiseeritud kõigi võimalike töö- e väljundoperatsioonide algoritmid. Juhtimine taandub siis vajalike algoritmide valikule ning nende järjestuse määramisele. Neid algoritme on võimalik realiseerida nn jäiga loogikaga, s.o vastavate loogikaskeemide abil, või mällu salvestatud e programmeeritava loogika abil. Viimasel juhul nimetatakse igale väljundoperatsioonile vastavat alamprogrammi mikroprogrammiks ja igale sisemisele elementaaroperatsioonile vastavat tehet mikrokäsuks. Mikroprogramme realiseerivat juhtseadme osa nimetatakse aga mikroprogrammjuhtimisega automaadiks. Seega on mikroprogrammjuhtimine üldise programmjuhtimise (mida nimetatakse vahel ka makroprogrammjuhtimiseks) alumiseks tasandiks. Kui makroprogrammid koostab peamiselt robotsüsteemi kasutaja, siis mikroprogrammid koostab enamasti seadet valmistav tehas ning need kuuluvad seadme riistvara (püsivara) juurde. Analoogiliselt on jaotatud programmjuhtimise ülesanded arvutis. Kasutajaprogramme töötleb protsessor, mille tööd korraldab juhtautomaat. Kui tehnoloogilist protsessi või robotsüsteemi juhib arvuti või mitmeraali juhtseade, siis võib programmjuhtimise hierarhiaastmete arv olla 3 ja enamgi.